DOCUMENT RESUME

ED 295 596 IR 013 247

AUTHOR Dixon, Victoria A.
TITLE An Investigation of Prior Sources of Difficulties in Learning University Computer Science.
PUB DATE 25 Jun 87
NOTE 36p.; Paper presented at the National Educational Computer Conference (Philadelphia, PA, June 25, 1987).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Academic Achievement; Cognitive Style; *Computer Science Education; Correlation; Higher Education; Hypothesis Testing; Mathematics Achievement; *Prior Learning
IDENTIFIERS Error Analysis (Programing)

ABSTRACT

The background of 197 university students from four introductory computer science classes at the State University of New York at Buffalo was investigated to determine factors contributing to academic success or failure in computer science. A background questionnaire, a test of computer science concepts (reading a simple program, symbol manipulation, language constructs, and algorithmic processing), and a learning style inventory were administered. It was hypothesized that the study would find: (1) significant differences in achievement for those who have started the study of computing prior to high school, in high school, and at the college level (rejected); (2) a significant positive correlation between mathematics scores and self-perceived prior mathematics achievement and current achievement (accepted); (3) positive correlations between achievement and both the number of semesters of computer studies (rejected) and the length of programs written (accepted); (4) significant differences in achievement between students with different initial programming language experience (rejected); (5) significant differences in learning style preference and achievement (rejected); and (6) significant differences in learning style preference for students selecting computer science as a major (accepted). Error analysis was performed on responses, and test scores were compared with sources of learning. It was concluded that the greatest positive impact on achievement derives from early exposure to computers and length of programs written. (98 references). (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED295596

AN INVESTIGATION OF PRIOR SOURCES OF DIFFICULTIES IN LEARNING UNIVERSITY COMPUTER SCIENCE

by Victoria A. Dixon

RESULTS OF THE STUDY:
Sources of Errors for University Computing Students

1 SUMMARY OF THE STUDY

The primary concern of this study was to investigate their backgrounds to determine what might have contributed to the academic success or failure of students interested in pursuing computer science.

To accomplish this task, a questionnaire was developed to obtain background information from each participating student. A test was also developed to test certain aspects that were judged by the investigator to measure performance of fundamental concepts that needed clarification and were necessary for a precise and complete understanding of computers and computer programming.

The test contains 17 questions: eleven questions required brief written protocol responses and six questions were multiple choice. Students were asked to give a brief explanation for their selection on the multiple choice questions to ascertain their understanding of the concept tested.

Another variable is the student's self-perception of the way he or she learns. This particular measure was included to determine if learning problems might be related to or corrected by a greater metacognitive awareness. The Learning Style Inventory developed by Kolb, McIntyre a

IR013247

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Victoria A. Dixon

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

others in 1976, revised by David A. Kolb in 1985 and published by McBer and Company in Massachusetts was used for this measure.

2 THE ACTUAL SAMPLE

2.1 The Population

Students from four specific computing class environments took part in the actual testing. The 197 students, all enrolled at The State University of New York at Buffalo, comprised 18 graduate students enrolled in LAI 676, Pascal Programming for Teachers; 19 students in EAS 101, Graphics with FORTRAN, a short course for interested high school students; 14 from the summer session of CS 113, the general introductory course in computer science at the University, and 146 students from two sections of the Fall CS 113 class.

2.2 Methods Of Analysis

The primary source for the statistical analysis of the data was Version 4.4 of the SPSSX software applications at Canisius College in Buffalo, New York. The analysis of variance (ANOVA), crosstabulation (CROSSTABS), and breakdown of the population (BREAKDOWN) procedures were used to find comparative results from the various groups and types studied. To find summative results of the types of errors and misconceptions, the CONDESCRIPTIVE and FREQUENCIES procedures were used. To supplement these statistical

analyses and to provide a cross check, hour-long protocol sessions were held and taped for further clarification and discussion of problems as they were perceived by two of the students. Protocol responses given with test answers by some of the students were also reviewed.

3 PRELIMINARY RESULTS

The 17 test questions were reviewed for measurement of reading a simple program (RASP), basic symbol manipulation (BSMP), language constructs (LNCS), and algorithmic processing (ALGP) as outlined in a test analysis. This classification resulted in a number of questions being included in more than one category; hence results of these measures are not independent. To get more information about particular types of misconceptions, the test questions were also divided into the eight specific error categories cited by Spohrer and Soloway (1985). Both condescriptive analyses and crosstabulations were done for each of these items with the total test score (TTST).

The results of the testing were mixed. This chapter provides information about results of the pilot study and of the final testing.

4 THE PILOT STUDY

A pilot study was undertaken during Spring, 1986 to test the effectiveness and clarity of the test questions as

well as the time required for each of the tasks.

A group of three individuals with some computer background took the preliminary test. The scores from this group were used specifically to test the content and design of the test. This initial study resulted in more explicit statements of variable names, the rewording of several questions and the replacement of three essay-type questions with multiple choice questions. Students also commented about the repetitious nature of the Learning Style Inventory. This led to a careful explanation of the reason for this repetition and scoring procedures of this instrument before administration in the actual testing.

5 THE FINAL STUDY

The explanations and discussion of the actual results are divided into two parts: the first part deals specifically with the five hypotheses; the second addressing the specific "high-frequency" errors listed by Spohrer and Soloway, 1985.

6 RESPONSES TO THE HYPOTHESES

Hypothesis 1

There are significant differences in achievement for those who have started the study of computing and about computers 1) prior to high school, 2) in senior high school, and 3) at the college level.

This hypothesis is not accepted. Though there are differences in the acievement of each group as noted below, they are not significant with the most favorable analysis producing only .10 < p < .15; Chi-square = 107.18, significance = 0.34.

This test information for students with different sources of initial computer science is given in Table 4.1.

7 TABLE 4.1 TEST RESULTS BY INITIAL EXPOSURE

GROUP	MEAN	STD DEV	N
ENTIRE POPULATION	33.3	7.3	197
ON OWN	34.7	9.0	20
GRD SCHL	36.2	5.2	18
HI SCHL	33.3	6.4	93
COLLEGE	32.0	8.2	66

For this analysis F = 1.84 and p = 0.14.

A second analysis of this data was undertaken. The different test scores (TTST) were recoded in percentile classifications (TST) ranging from 1 for scores greater than 90% down to 5 for those less than 50%. With the recoding of the test scores into five percentile levels, F = 0.936 with the corresponding p = 0.44. This clearly confirms rejection of hypothesis 1. This data is displayed in table 4.2.

The general key to the boxes here and in all of the crosstabulations that follow are:

COUNT, the actual number of students
COL PCT, the percentage of students in the column

TOT PCT, the total percentage of all students in the particular category

7.1 TABLE 4.2 TEST ACHIEVEMENT BY GROUP

GROUP	TST > 90%	75-89%	60-74%	50-59%	< 50%	ROW TOTAL
ON OWN	2 33.3 1.0	6 11.5 3.1	6 8.1 3.1	2 5.4 1.0	4 14.8 2.0	20 10.2
GRD SCHL		10 19.2 5.1	5 6.8 2.6	2 5.4 1.0	1 3.7 .5	18 9.2
HI SCHL	1 16.7 .5	21 40.4 10.7	40 54.1 20.4	19 51.4 9.7	11 40.7 5.6	92 46.9
COLLEGE	3 50.0 1.5	15 28.8 7.7	23 31.1 11.7	14 37.8 7.1	11 40.7 5.6	66 33.7
COLUMN TOTAL	6 3.1	52 26.5	74 37.8	37 18.9	27 13.8	196 100.0

This hypothesis states that there will be a significant difference in test scores for those who have started the study of computing and about computers 1) prior to high school (GRD SCHL), 2) in senior high school (HI SCHL), and 3) at the college level (COLLEGE). Because of the significant number of students who began their studies independently or "on their own", a fourth classification (ON OWN) was included in the data entry.

Note that there is a wider variation in test scores for students that learned about computing independently and students who began their work in college. The least variation was for those students who started in grade school

It should also be noted that more than 50% of those who started their study of computing in grade school (10 out of 18) scored at least 75% of the test questions correct, though none of them achieved 90% or better. In contrast, only 40 students of the 158 or slightly more than 25% who started studies in high school or college achieved the same rank.

Hypothesis 2

> There is a significant positive correlation between Scholastic Aptitude math scores (SATMTH) AND self-perceived prior mathematics achievement (MTHRAT) and current (TTST) achievement.

This hypothesis is accepted. It was investigated in two ways. First, test scores (TTST) were measured against self estimation of mathematics achievement (MTHRAT) from item No. 7 of the questionnaire. The Pearson correlatiom was used and produced $r = .2154$ for $N = 197$, $p = .001$.

Table 4.3 shows test scores and mathematics self-rating. It can be seen that the six who classified themselves as excellent did obtain a score with greater than 90%. The one person who self-classified as poor received less than 50%.

7.2 TABLE 4.3 MATH SELF ANALYSIS AND ACHIEVEMENT

MTHRAT	TST >90%	75-89%	60-74%	50-59%	<50%	ROW TOTALS
POOR					1 3.7 .5	1 .5
FAIR		5 9.6 2.6	5 6.8 2.6	4 10.8 2.0	2 7.4 1.0	16 8.2
GOOD		17 32.7 8.7	29 39.2 14.8	19 51.4 9.7	16 59.3 8.2	81 41.3
EXCEL	6 100.0 3.1	30 57.7 15.3	40 54.1 20.4	14 37.8 7.1	8 29.6 4.1	98 50.0
COLUMN TOTALS	6 3.1	52 26.5	74 37.8	37 18.9	27 13.8	196 100.0

Second, test scores were measured against reported Scholastic Achievement Test scores (SATMTH). The Pearson correlation produced r = .3507, N = 149 AND P = .000. These highly significant results clearly confirm the hypothesis.

A further analyis was done to determine relationships between student's self-perception and the actual SAT scores that were reported. These analysis results were also highly significant with $p < .001$. This indicates that the students were generally accurate in self perception and lends support to their own representations. It can be noted, however, that there are some interesting contradictions; some students who rated themselves as good or excellent received less than 50% on the test.

7.3 TABLE 4.4 SAT SCORES AND TEST ACHIEVEMENT

SATMTH BY TST / SATMTH	>90%	75-89%	60-74%	50-59%	< 50%	ROW TOTALS
< 400					1 6.3 .7	1 .7
400-499				3 10.3 2.1	1 6.3 .7	4 2.8
500-550		1 2.9 .7	8 14.3 5.6	8 27.6 5.6	7 43.8 4.9	24 16.9
551-620		10 28.6 7.0	24 42.9 16.9	10 34.5 7.0	5 31.3 3.5	49 34.5
> 620	6 100.0 4.2	24 68.6 16.9	24 42.9 16.9	8 27.6 5.6	2 12.5 1.4	64 45.1
COLUMN TOTAL	6 4.2	35 24.6	56 39.4	29 20.4	16 11.3	142 100.0

NUMBER OF MISSING OBSERVATIONS = 55
r = .3507, N = 149 AND P = .000.

These results show there is a highly significant correlation between success in computer science and ability in mathematics, a result that confirms the research reported by Ralston (1980, 1981), Konvalina and others (1983), Werth (1986) and Dey and others (1986).

Hypothesis 3

There are positive correlations between achievement and a) the number of semesters of computer studies, and b) the length of programs written.

This hypothesis is rejected for a) semesters of study and accepted for b) the number of lines in program code. The Pearson correlation of achievement by the number of semesters (TOTSEM) yielded r = .0570, N = 197 WITH p = .213; this same statistical procedure yielded = .3188 for N = 197 and p < .001 for number of lines of code written (MAXLINE) with the total test score.

The variable used for the number of lines in a program (MAXLINE) is the maximum number of lines reported in question 10 of the questionnaire for any of the five program languages considered. The variable for the number of semesters of study (TOTSEM) was found by totaling the number of semesters of study in each language and subtracting one to discount the current semester of study.

Each of these variables were categorized into five subdivisions. An analysis of test achievement and the number of lines of code written is given in Tables 4.5 and 4.6.

7.4 TABLE 4.5 LENGTH OF PROGRAMS AND TEST SCORES

COUNT / MAXLINE	TST >90%	75-89%	60-74%	50-59%	<50%	ROW TOTAL
0-50		4	14	6	11	35
51-75		3	1	2		6
76-120	1	8	13	9	9	40
121-200		10	10	3	1	24
>500	5	27	36	17	6	91
COLUMN TOTAL	6	52	74	37	27	196
COLUMN TOTAL %	3.1	26.5	37.8	18.9	13.8	100.0

The relationship of the test scores with the total number of semesters (TOTSEM) of study is given in table 4.6.

7.5 TABLE 4.6 SEMESTERS OF STUDY AND TEST SCORES

TOTSEM Semesters of study BY TST

COUNT / TOTSEM	TST >90%	75-89%	60-74%	50-59%	<50%	ROW TOTAL
1-2	1	15	25	15	13	69
3-4	3	18	30	13	12	76
> 4	2	19	19	9	2	51
COLUMN TOTAL	6	52	74	37	27	196
COLUMN TOTAL	3.1	26.5	37.8	18.9	13.8	100.0

This lends further support to the hypothesis that longer programs presumably requiring more coding with greater emphasis on logical design and algorithmic applications do contribute to a better understanding of computing concepts, but it calls into question the results for prior semesters of experience. This also confirms the argument by Luehrmann (1982) discussed in Chapter 1 that actually doing

computing is a basic intellectual skill and does contribute to student learning.

This result would seem to indicate that students often did not get actively involved in in-depth algorithmic development or processing and hence did not develop the necessary background required of computer science majors.

It should be noted that there was an obvious misunderstanding by subjects of what was understood as data for the first semester course. From observation of the raw data, a number of students classified learning on their own equivalent to one or two semesters of formal study.

The relationship of a person's innate learning procedures and interest in such topics as computer science and their performance on the test questions is now given. This investigation will examine the role of one's self- perception of the way he or she learns best and the test performance.

Hypothesis 4

> There are significant differences in achievement (TTST) between those students who have different initial programming language experience (LNG).

Hypothesis 4 is not accepted with $p > .10$. This aspect of the study showed that prior programming language learning experience did contribute to the student's achievement, however these results must be viewed with caution. In this tally, only one student had initial programming experience

in COBOL and only two students in LOGO. In another analysis, it was confirmed that there were 15 students with COBOL experience and 20 with LOGO experience. A crosstabulation was done between initial languages learned and test achievement with the following results:

7.6 TABLE 4.7 Initial Languages And Test Achievement

COUNT / ROW PCT / COL PCT	BASIC	FORTRAN	PASCAL	LOGO	COBOL	ROW TOTAL
TST						
>90%	1 33.3 2.0		1 33.3 4.5		1 33.3 100.0	3 3.5
75-89%	12 54.5 24.5	5 22.7 41.7	5 22.7 22.7			22 25.6
60-74%	23 65.7 46.9	3 8.6 25.0	7 20.0 31.8	2 5.7 100.0		35 40.7
50-59%	8 53.3 16.3	3 20.0 25.0	4 26.7 18.2			15 17.4
<50%	5 45.5 10.2	1 9.1 8.3	5 45.5 22.7			11 12.8
COLUMN TOTAL	49 57.0	12 14.0	22 25.6	2 2.3	1 1.2	86 100.0

F = 0.0022
SIGNIFICANCE = .035

There are variations between the different languages, but they are not significant as noted below. The large number of missing values are due in part to those who chose to omit the initial grade when programming was learned and also the significant number who learned "on their own".

	Mean	No
TOTAL POPULATION	33.6	87
BASIC	33.9	50
FORTRAN	34.7	12
PASCAL	31.5	22
LOGO	36.5	2
COBOL	46.0	1

110 CASES (55.8 PCT) WERE MISSING.

A ONEWAY analysis of variance between achievement (TTST) and the various language groups was also done to determine if there was any significant variations between the means for the different language groups. The results for an analysis of variance for BASIC, FORTRAN, and Pascal showed F = 1.634 with significance = 0 =.20. When the three persons in LOGO and COBOL were included, there was a significant difference; this result was discounted because of the small number of students. The procedure CONTRASTS was used to determine if there was any significant differences between achievement and the groups for BASIC, FORTRAN, and Pascal. The differences in the achieveement (TTST) and the three languages were not significant.

It was assumed that those who did not have the discipline of a top down structured background might not achieve as well on the test. Contrary to expectations, it is interesting to note that one student each in BASIC, Pascal, and LOGO each achieved a score > 90%. The contrast showed p > .26 for the comparison of FORTRAN with Pascal, p > .73 for BASIC with FORTRAN and P > .28 for BASIC with Pascal. The other comparisons were ignored because of the negligent

number of subjects involved.

Hypothesis 5

There will be significant differences in learning style preference (TYPE) and achievement (TTST).

This hypothesis is not accepted. The probability value was $p > .80$. The crosstabulation procedure yielde Chi square = 7.29 with significance $p = 0.8374$.

The variation in test scores between groups is small as shown below:

TOTAL POPULATION	33.22 (196)		
Diverger	Assimilator	Converger	Accommodator
33.42	32.52	32.45	33.59
(55)	(25)	(31)	(85)

The fifth hypothesis addresses the self-perception of a student as a learner. Each student was asked to complete the Kolb Learning Style Inventory as specified in Chapter 2 and Chapter 3. In this inventory, students were asked to complete twelve sentences, each with four endings. They were asked to rank the endings of each sentence according to how well they thought it fit with how they went about learning. Each sentence ending was ranked from "1" that seemed the least like the way they would learn up to a "4" for how they learned best.

Twelve individuals did not follow the learning style inventory instructions correctly. The test results for these students are not included in the learning style analysis. Table 4.8 given below shows the relationship between the self-perception scores, TYPE, and the recoded achievement, TST.

These results were not significant with F = 0.27 for which the corresponding p = .84; the analysis did show, however, that students, 26 accommodators and 15 divergers which represented more than 20% of the students in the study, did achieve at least 75% correct on the test.

7.7 TABLE 4.8 TST (TEST ACHIEVEMENT) BY TYPE

COUNT / COL PCT / TOT PCT	TYPE DIVRGER	ASMLATR	CNVRGER	ACOMDTR	ROW TOTAL
TST					
> 90 %	2 3.6 1.0			4 4.8 2.1	6 3.1
75-89%	13 23.6 6.7	8 32.0 4.1	8 25.8 4.1	22 26.2 11.3	51 26.2
65-74%	22 40.0 11.3	7 28.0 3.6	14 45.2 7.2	31 36.9 15.9	74 37.9
50-64%	12 21.8 6.2	6 24.0 3.1	3 9.7 1.5	16 19.0 8.2	37 19.0
< 50%	6 10.9 3.1	4 16.0 2.1	6 19.4 3.1	11 13.1 5.6	27 13.8
COLUMN TOTAL	55 28.2	25 12.8	31 15.9	84 43.1	195 100.0

From these results, it appears that the actual preferred style of learning has no significant relation to the test scores. This was verified crosstabulation which showed Chi Square =7.3 and significance = .83. At the same time, it can be seen that six students who achieved > 90% (3.1% of the sample) were either Diverger or Accommodator styles, both of which prefer concrete experience. There were no Assimilators or Convergers in this group. It can also be noted that more than 20% of the students tested who received a score greater than or equal to 70% were of these same types of preferred learning.

Hypothesis 6

> There are significant differences in learning style preference (TYPE) for students selecting computer science as a major.

Hypothesis 6 is accepted. Nearly 74% or 54 of the 73 students who responded correctly to the Learning Style Inventory and who selected computer science or computer science related majors have either Diverger or Accommodater preferred learning styles. The result of an analysis of variance is detailed below in table 4.9 and has r = .1395 with p = .029 with all six MJR descriptions.

The sixth hypothesis states that students who select computer science as a chosen major will have different learning style preferences from other students. This premise is made on the observation and descriptions of the various styles by Kolb (1985). Students who prefer getting things done and risk taking (Accommodator) and have

imaginative ability (Diverger) will be interested in the dynamism of computer science and new technology.

To test this hypothesis, students were asked to list their intended major. The resulting information was tallied under the variable MAJOR which yielded more than 25 variations. To make the data less cumbersome for analysis of variance, these variations were tabulated into 16 different areas. For cross tabulations and analysis of variance, the 16 variations were recoded into six classes in variable MJR. All students who specified computer science, computer science education or management information systems were classified as 1 in the MJR ranking. The variable name CompSci stands for computer science, MtScEd for mathematics and science education; MtStLw for mathematics, statistics and law; Science for physics, biology, micro-biology, and chemistry; EngCom for English and communications; and OthUnd for other and undecided. Law was included with mathematics because of the general need for logic and precision.

The comparatively large number of missing observations can be attributed to the exclusion of those students who did not follow directions correctly in selecting priorities in the ways they tended to learn. Data for this group were entered under a special category code for achievement results and other pertinent information.

The original classifications of MAJOR interests were recoded into the six group classifications, MJR, as specified above. Cross tabulations were done for TYPE (learning style) and MJR (major) with the following results:

8 TABLE 4.9 CROSSTABS TABLES = MJR (MAJOR) BY TYPE

COUNT / COL PCT / TOT PCT	TYPE				ROW
MJR	DIVRGER	ASMLATR	CNVRGER	ACOMDTR	TOTAL
CompSci	25 58.1 15.0	8 36.8 4.8	11 40.7 6.6	29 38.7 17.4	73 43.7
MtScEd	4 9.3 2.4	1 4.5 .6	1 3.7 .6	6 8.0 3.6	12 7.2
MtStLw	3 7.0 1.8	1 4.5 .6	1 3.7 .6	6 8.0 3.6	11 6.6
Science	4 9.3 2.4	3 13.6 1.8	4 14.8 2.4	4 5.3 2.4	15 9.0
EngCom	2 4.7 1.2	4 18.2 2.4	2 7.4 1.2	10 13.3 6.0	18 10.8
OthUnd	5 11.6 3.0	5 22.7 3.0	8 29.6 4.8	20 26.7 12.0	38 22.8
COLUMN TOTAL	43 25.7	22 13.2	27 16.2	75 44.9	167 100.0

NUMBER OF MISSING OBSERVATIONS = 30
Chi-square = 13.39 with significance = 0.57.

The Pearson correlation gave r = .1365, N = 194, and p = .029.

This indicates that students with certain types of learning styles and learning characteristics are more prone to fields such as computer science. This information can be used to determine possible sources of difficulties in learning computer science as well as weaknesses to be compensated.

8.1 Programming Concepts Addressed

As noted in the test question analysis, four specific concepts were tested: Reading a Simple Program (RASP), Basic symbol manipulation (BSMP), Language Constructs (LNCS), and Algorithmic processing (ALGP). The test questions were classified in each category depending on whether the concept was essential for the correct answer. The category on basic symbol manipulation included more questions than the others, since this is considered to be essential to proper understanding of computer concepts (Perlis, 1982) and includes many aspects of programming. In the analysis breakdown, four questions provided information on reading a simple program, 16 of the 17 test questions included basic symbol manipulation, eight were considered to measure perception of language constructs and eleven were included in testing algorithmic processing. Since each question was given a maximum score of three and a minimum of zero, the range for each is 0-12 for RASP, 0-48 for BSMP, 0-24 for LNCS, and 0-33 for ALGP.

To find the range of answers correct and the mean for each concept tested, a condescriptive statistical procedure was used. The percent correct was computed by dividing the mean into the total value possible. To supplement data scores provided by this measure, two hour-long verbal protocols were conducted, taped and transcribed. The complete transactions of these interviews can be found in Appendix A.

Though not every question on the test was included in the verbal protocol transactions, each of the four concepts tested and each of the eight error types measured were addressed. This was verified by checking the questions included in the interview and the test analysis.

The following results were obtained:

8.2 TABLE 4.10 CONCEPTS AND TEST MEANS

Concept	Possible	Obtained	Mean	Std Dev	%Correct
RASP	0 - 12	0 - 12	8.79	2.74	73.3
BSMP	0 - 48	11 - 47	31.49	7.01	65.6
LNCS	0 - 24	4 - 24	16.39	4.03	68.3
ALGP	0 - 33	7 - 33	22.00	4.97	67.7

The category with the largest standard deviation is basic symbol manipulation (BSMP); this category always has the lowest percent correct with less than two-thirds correct. The percent correct for the group was computed by dividing the total possible by the mean achieved. This

would seem to indicate that a greater emphasis should be placed on the notion of symbol nanipulation in the introductory computer science course. This conclusion is supported by information from the protocol interviews (see Appendix A).

A further evaluation was done for those students who stated computer science as their chosen major. The means for each of these categories were only slightly higher than those given above for the entire group. The slightly higher means were consistent for the four concepts specified. See Table 4.10 giving means only for the entire group.

8.3 Specific Errors

To determine if any specific error was more prevalent in the particular test items, a further subdivision of the test questions was made. The eight most frequently made errors specified by Spohrer and Soloway, (1985) were analyzed for these test questions.

To obtain data for this aspect of the study, the CONDESCRIPTIVE statistical procedure was again used. The eight errors analyzed are the inappropriate use of knowledge in writing step-by-step procedural specifications (INAPP), the expectation or misinterpretation of a READ statement as a declaration statement (MISIN), a false assumption about incrementing a counter, (OFFBY1) the misplacement of or failure to initialize a counter variable (UNVAR), the omis-

sion of the necessary specification to initial a total summation variable to zero (ZEROM), the failure to clarify or understand the organization of the algorithm (ORGCL), the use of a wrong formula (WRNGF), and interchanging OR for AND (ORAND).

The test questions were analyzed for each of these possible errors as indicated in table 4.11. The inappropriate use of knowledge (INAPP) was the most prevalent in the test questions with 16 questions included. The "off by one" (OFFBY1) and "lack of organization" (ORGCL) were the next highest categories with four and five questions respectively. The wrong formula (WRNGF) was in three of the questions; the remainder of the categories included two each: misinterpretation of the read statement, zero omitted, uninitialized variable and the use of OR for AND (MISIN, ZEROM, UNVAR, ORAND). Though the number of some of these questions could not be used to establish significant results, they are included to show how these particular students responded in each category.

In general, the means were slightly greater than the computed mean. The mean and percent correct for organization clarification (96.1) was much above the percent correct for any of the other categories. This result suggests the possibility of an overemphasis on structure rather than on attention to initial required specifications. The results are given in table 4.11.

8.4 TABLE 4.11 ERRORS AND TEST MEANS

Concept	Possible	Obtained	Mean	Std Dev	%Correct
INAPP	0 - 48	11 - 47	31.71	6.79	66.1
MISIN	0 - 6	0 - 6	4.06	1.51	67.7
OFFBY1	0 - 12	0 - 11	7.90	2.53	65.8
UNVAR	0 - 6	0 - 6	3.31	1.68	55.2
ZEROM	0 - 6	0 - 6	3.49	1.59	58.2
ORGCL	0 - 15	6 - 14	14.42	3.02	96.1
WRNGF	0 - 9	0 - 9	5.19	1.81	57.7
ORAND	0 - 6	0 - 6	3.27	1.35	54.5

For the four variables with just two questions, there was a noted difference in the proper use and interpretation of the READ statement (MISIN) with a mean of 4.06 while the other three, the uninitialized variable (UNVAR), the omission of zero (ZEROM) and the misuse of the OR and AND statements (ORAND) had lower means of 3.31, 3.49, and 3.27 respectively. The failure to initialize a variable for counting purposes, indicated in the UNVAR, ZEROM variables are evident in student responses in the protocol sessions for question 11 on page A-6. There was an overlapping of at least one test question in each of the categories. The greatest impact of this overlapping would be in the results for OFFBY1 and ZEROM; the two questions for zero omitted (ZEROM) are included in the four questions for "off by one" (OFFBY1).

8.5 Sources Of Learning

The relationship of a person's test score is now compared with the source of the individual's learning about computer science. To determine if the original source of learning had any implication on test scores for the individuals in this study, a crosstabulation was done between the test scores and the stated source of learning. The following results were obtained:

8.6 TABLE 4.12 Test Scores And Source Of Learning

COL PCT TOT PCT TST	IN THIS COURSE	IN SCHOOL	ON MY OWN	ROW TOTAL
>90 %		3 2.2 1.5	3 7.3 1.5	6 3.1
75-89	1 7.1 .5	38 27.3 19.6	13 31.7 6.7	52 26.8
60-74	3 21.4 1.5	55 39.6 28.4	14 34.1 7.2	72 37.1
50-59	2 14.3 1.0	28 20.1 14.4	7 17.1 3.6	37 19.1
<50 %	8 57.1 4.1	15 10.8 7.7	4 9.8 2.1	27 13.9
COLUMN TOTAL	14 7.2	139 71.6	41 21.1	194 100.0

Chi-square for GROUP with TTST = 107.18, significance = .34.

This study is slightly different from that in table 4.2. This chart shows the achievement based on whether or not students received background knowledge in a structured situation prior to this course.

Though the numbers involved in this study for those with their first experience at computing coming in this course (14) are limited, the students who did not have any background in computer programming or computer science did appear to have a noted disadvantage in the beginning course. Nearly seventy percent of those students (10) scored less than 60% on the test questions.

It has been said (Gorsline, 1981) and by educators at various conferences: ACM, SIGCSE (1981, 1982, 1985), WCCE (1986) that many students who have learned about computers on their own or from friends do not have the proper perspectives and background knowledge to grasp some of the requirements for a correct application of certain concepts. This has been contradicted in this study. Nearly twelve percent of the students (23) who started computing in high school or in this course received less than 50% while slightly more than two percent (4) students who learned on their own were in this category.

From the analyses given, the greatest positive impact on achievement seems to derive from early exposure to computers and the length of programs written. Learning Style Inventories are useful, especially for college age students.

They can be encouraged to recognize weaknesses in their learning approaches and consciously work on the more positive aspects of their learning type.

REFERENCES

Anderson, R. The Minnesota Educational Computing Consortium (MECC) Project on Computer Literacy Instructional Modules (CLIM). In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), Computer Literacy. New York: Academic Press, 1982.

Anderson, R. E. Monitoring the effect of computers on education. In TOPICS in Computer Education: National Educational Computer Policy Alternatives. SIGCUE Bulletin, Association for Computing Machinery, 1986, 18, 16-21.

Anderson, R. & Klassen, D. A Conceptual Framework for Developing Computer Literacy Instruction. AEDS Journal, 1981, 14(3), 128-150.

Anderson, R. E. Comprehensive Computer Literacy for College Students. Paper for Minnesota Center for Social Research. Minneapolis: July, 1983.

Anderson, R., Klassen, D. & Johnson, D. In defense of a comprehensive view of computer literacy--a reply to Luehrmann. The Mathematics Teacher, 1981, 74, 687-690.

Austing, R. H., B. H. Barnes, D. T. Bonnette, G. L. Engel, & G. Stokes (Eds.). Curriculum '78: Recommendations for the Undergraduate Program in Computer Science. Communications of ACM, 1979, 22, 147-166.

Bayman, Piraye, & Mayer, Richard E. A Diagnosis of Beginning Programmers' Misconceptions of BASIC Programming Statements. Communications of ACM, 1983, 26, 677-679.

Bloom, B. Taxonomy of Educational Objectives. Handbook I: Cognitive Objectives. New York: Longmans, 1956.

Bloom, B. S., Hastings, J. T., & Madaus, G. F. Handbook on Formative and Summative Evaluation of Student Learning. New York: McGraw Hill, 1971.

Bobrow, D. G. & Collins, A. (Eds.) Representation and Understanding: Studies in Cognitive Science. New York: Academic Press, 1975.

Bonar, J. & Soloway, E. Uncovering principles of novice programming. NSF Grant SED-81-12403, November, 1982.

Bonar, J. & Soloway, E. Pre-Programming knowledge: A major source of misconceptions in novice programmers. Human-Computer Interaction, Fall, 1985.

Brown, J. S. III. Cognitive Research and Solving Problems Using the Computer. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), Computer Literacy. New York: Academic Press, 1982.

California Department of Education, California Assessment Program. Sacramento: California Department of Education, 1983.

Campbell, Donald T. & Stanley, Julian C. Experimental and Quasi-Experimental Designs for Research. Chicago: Rand McNally, 1963.

Carpenter, T. P., Lindquist, M. M., Matthews, W. & Silver, Edward A. Results of the Third NAEP Mathematics Assessment: Secondary School. The Mathematics Teacher, 1983, 76(9), 652-659.

Center for Educational Intelligence. Executive Suummary: Twelfth grade achievement in computer studies. California Department of Education, 1983.

Clement, J. Mapping a student's causal conceptions from a problem-solving protocol. In J. Lochhead & J. Clement (Eds.), Cognitive Process Instruction: Research on Teaching Thinking Skills. Philadelphia: Franklin Institute Press, 1979.

Craig, R. T. Implications of Recent Research in the "Computational Paradigm". Paper presented at the Annual Meeting of the International Communication Association. Chicago, Illinois: 25-29 April 1978.

Dalbey, J. and Linn, M. Cognitive consequences of programming: Augmentations to basic instruction. Journal of Educational Computing Research, 1986, 2(1), 75-93.

D'Amour, G. Problem-solving strategies and the epistemology of science. In J. Lochhead & J. Clement (Eds.), Cognitive Process Instruction: Research on Teaching Thinking Skills. Philadelphia: Franklin Institute Press, 1979.

Dey, Sutken, & Mand, L. R. Effects of mathematics preparation and prior language exposure on perceived performance in introductory computer science courses. SIGCSE Computer Science Education ACM Bulletin, 1986, 18(1), 144-148. presented at 17th Annual SIGCSE Conference. Dayton: February 7, 1986.

Dijkstra, E. W. Go To statement considered harmful. Communications of ACM, 11, 147-148.

Dresher, B. E. & Hornstein, N. On some supposed contributions of artificial intelligence to the scientific study of language. Cognition, 1976, 4, 321-398.

DuBoulay, Benedict. Some difficulties of learning to program. Journal of Educational Computing Research, 1986, 2(1), 57-73.

Ewing, David W. Discovering Your Problem-Solving Style. Psychology Today, December, 1977, 69-74.

Ferrell, B. A factor analytic comparison of four learning-styles instruments. Journal of Educational Psychology, 1983, 75(1), 33-39.

Fetler, Mark. Computer Literacy of California Sixth and Twelfth Grade Students. Division of Planning Evaluation and Research. Sacramento: California Department of Education, November, 1983.

Foley, J. (Session Chairman). Teaching Human Factors in Computer Systems. SIGCSE Computer Science Education ACM Bulletin, 1982, 14(3), 31-44.

Garofalo, J. & Lester, F. K. Metacognition, cognitive monitoring, and mathematical performance. Journal for Research in Mathematics Education, 1985, 16(3), 163-176.

Gear, C. W. Introduction to Computer Science. Chicago: Science Research Associates, 1973.

Geller, Lester M. Reliability of the learning style inventory. Psychological Reports, 1979, 44, 555-561.

Ginsberg, H., N. Kossan, R. Schwartz, & D. Swanson. Protocol Methods in Research on Mathematical Thinking. In Development of Mathematical Thinking. New York: Academic Press, 1983.

Goldschlager, L. & Lister, A. Computer Science: A Modern Introduction. London: Prentice-Hall International, 1982.

Gorsline, G. W. Articulation: easing the high school transition, SIGCSE Computer Science Education ACM Bulletin, 1982, 14(3), 24-29.

Greenes, C. The computer in mathematics education. The Mathematics Teacher, 1981, 74, 588-589.

Greeno, J. Cognitive objectives of instruction: Theory of knowledge for solving problems and answering questions. In D. Klahr (Ed.), Cognition and Instruction.

Hillsdale, NJ: Lawrence Erlbaum Associates, 1976.

Gries, D. The science of programming. Paper presented at ACM-CSC 82 Computer Science Conference. Indianapolis, IN: 11 February 1982.

Gronlund, N. E. Readings in Measurement and Evaluation. New York: Macmillan, 1968.

Hallworth, H. J. and Brebner, A. Computer Assisted Instruction in Schools. Calgary, Alberta: University of Calgary, 1980.

Hunt, D. E. Person-environment interaction: A challenge found wanting before it was tried. Review of Educational Research, Spring, 1975, 45(2), 209-230.

Hunt, D. E. Learning style and student needs: An introduction to conceptual level. In Student Learning Styles. Reston, VA: National Association of Secondary School Principles, 1979, 27-38.

Johnson, J. Four expert opinions. Datamation, 1982, 28(13), 143-154.

Joni, S. A. & Soloway, E. But my program runs! Discourse rules for novice programmers. Journal of Educational Computing Research, 1986, 2(1), 95-126.

Khatena, J. & Torrance, E. Manual for Khatena-Torrance Creative Perception Inventory. Chicago: Stoelting Co., 1976.

Klassen, D., R. Anderson, T. Hansen, and D. Johnson. A Study of Computer Use and Literacy in Science Education, Final Report. St. Paul: Minnesota Educational Computing Consortium, 1980.

Knuth, Donald. Algorithmic thinking and mathematical thinking. American Mathematical Monthly. 1985, 92(3), 172-181.

Kolb, D. A. L S I Learning Style Inventory. Boston: McBer and Co., 1985.

Kolb, D. A., Rubin, I. M., & McIntyre, J. M. Organizational Psychology an experimential approach (3rd ed.) Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979.

Konvalina, J., Stanley A. Wileman, Larry J. Stephens. Math Proficiency: A key to success for computer science students. Communications of ACM, 26 (May, 1983), 377-382.

Lochhead, J. & Clement, J. (Eds.) Cognitive Process Instruction: Research on Teaching Thinking Skills. Philadelphia: Franklin Institute Press, 1979.

Luehrmann, A. Computer literacy (in reader reflections). The Mathematics Teacher, 1982, 75, 281-283.

MacNeil-Lehrer Report. Electronic Teaching. New York: Educational Broadcasting Corporation, Transcript 1769, 8 July 1982.

MacNeil-Lehrer Report. Artificial Intelligence. New York: Educational Broadcasting Corporation, Transcript 1975, 22 April 1983.

Mayer, R. E. Different problem solving competencies established in learning computer programming with and without meaningful models. Journal of Educational Psychology, 1975, 67, 725-734.

Mayer, R. A psychology of learning BASIC. Communications of ACM, 1979, 22, 589-593.

Mayer, R. The psychology of learning computer programming by novices. ACM Computing Surveys, 1981, 13, 121-141.

Mayer, R. Psychology of calculator languages: A framework for describing differences in users' knowledge. Communications of ACM, 1981, 24, 511-520.

Messick, Samuel. Research Report: Cognitive Styles in Educational Practice. Educational Testing Service, Princeton, NJ: June, 1982.

Minnesota Educational Computing Consortium. Computer Awareness and Literacy: An Empirical Assessment, (Publication 503), St. Paul, MN: 1982.

Moran, T. An applied psychology of the user. ACM Computing Surveys, 1981, 13, 1-11.

Nathan, J. Learning from the past: Guidelines for the future. In TOPICS in Computer Education: National Educational Computer Policy Alternatives. SIGCUE Bulletin, Association for Computing Machinery, 1986, 18, 130-143.

National Assessment of Educational Progress. The Third National Mathematics Assessment: Results, Trends and Issues (Advance Copy). Denver: April, 1983.

National Institute of Education. Science & Technology and Education: Cognitive Science, Organizational Science, Information Technology. Reprint from The Five-Year Outlook: Problems, Opportunities and Constraints in

Science and Technology. Washington: National Science Foundation, 1980.

Newell, A. & Simon, H. Computer science as empirical inquiry. Communications of ACM, 1976, 19, 113-126.

Newell, A. & Simon, H. Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall, 1972.

Nunamaker, J. F., Couger, J. D., & Davis, G. B. (Eds.). Information systems curriculum recommendations for the 80s: Undergraduate and graduate programs. Communications of ACM, 1982, 25(11), 781-805.

Papert, S. Mindstorms: Children, Computing, and Powerful Ideas. New York: Basic Books, 1980.

Pea, Roy D. Language-independent conceptual "bugs" in novice programming. Journal of Educational Computing Research, 1986, 2(1), 25-36.

Perlis, A. What is the computer science for which we are now preparing our students? Paper presented at ACM-CSC 82 Computer Science Conference. Indianapolis, IN: 11 February 1982.

Poirot, J., R. Taylor & J. Powell. Teacher education. TOPICS: Computer Education for Elementary and Secondary Schools. New York: Association for Computing Machinery, 1981, 18-27.

Ralston, A. Computer science, mathematics, and the undergraduate curriculum in both. The American Mathematical Monthly, 1981, 88, 472-485.

Ralston, A. & Shaw, M. Curriculum '78--is computer science really that unmathematical? Communications of ACM, 1980, 23, 67-70.

Resnick, Lauren B. Mathematics and science learning: A new conception. Science, 1983, 220, 477-478.

Rosnick, Peter. Some misconceptions concerning the concept of variable. The Mathematics Teacher, 1982, 75, 418-421.

Rosnick, Peter & John Clement. Learning without understanding: The effect of tutoring strategies on algebra misconceptions. Journal of Mathematical Behavior, 1982, 3(1), 3-24.

Sammet, J. E. & Ralston. A. (Eds.). The new (1982) ACM Computing Reviews classification system--final version. Communications of ACM, 1982, 25, 13-25.

Schank, R. C. & Abelson. Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures. Hillsdale, N. J.: Lawrence Erlbaum Associates, 1977.

Seidel, R. J. &, R. E. Anderson & B. Hunter (Eds.). Computer Literacy. Reston, VA: 1980.

Skinner, B. Thinking machines. Science, 1958, 128, 969-977.

Sleeman, D., Putnam, R., Baxter, J. and Kuspa, L. Pascal and high school students: A study of errors. Journal of Educational Computing Research, 1986, 2(1), 5-23.

Solow, D. How to teach abstract mathematical thought processes. Paper presented at the National Council of Teachers of Mathematics annual meeting. Toronto: 14 April 1982.

Soloway, E., Bonar, J. & Ehrlich, K. Cognitive Strategies and Looping Constructs. Communications of ACM, 1983. 26, 853-860.

Soloway, E., Ehrlich, K., Bonar, J., & Greenspan, J. What do novices know about programming? In A. Badre & B. Schneiderman (Eds.), Directions in Human/Computer Interaction. Norwood, NJ: Ablex Publishing Corporation, 1982.

Spoher, J. C. & Soloway, E. The high frequency bugs in novice programs: Evaluating some programming folk wisdom. Cognition and Programming Project. New Haven: Yale University Press, 1985.

Taylor, R. Arguments for computing in education. TOPICS: Computer education for elementary and secondary schools. New York: Association for Computing Machinery, 1981, 7-9.

Taylor, R. Definitions and Requirements for Computer Literacy. In R. J. Seidel, R. E. Anderson, & B. Hunter (Eds.), Computer Literacy. New York: Academic Press, 1982.

TOPICS: Computer education for elementary and secondary schools. K. Billings (final Ed.). Joint Education Board, SIGCSE, SIGCUE. New York: Association for Computing Machinery, 1981.

TOPICS: Computer education for colleges of education, Jean B. Rogers (Ed.) Joint Education Board, SIGCSE, SIGCUE. New York: Association for Computing Machinery, 1983.

Tukey, J. W. Exploratory Data Analysis. Don Mills, ONT: Addison Wesley, 1977.

Turkle, Sherry. The Second Self: Computers and the Human Spirit. New York: Simon & Schuster, 1984.

van der Veer, G. C. & J. van de Wolde. Psychological aspects of problem solving with the help of computer languages. Computers and Education, 1982, 6, 229-234.

von Neumann, John. Theory of Self-Reproducing Automata, ed. Arthur W. Burns. Urbana, IL: University of Illinois Press, 1966.

Werth, L. W. Predicting student performance in a beginning computer science class. SIGCSE Computer Science Education ACM Bulletin, 1986, 18(1), 138-143.

Weizenbaum, J. Computer Power and Human Reason: From Judgement to Calculation. San Francisco: W. H. Freeman, 1976.

Wessels, M. G. Cognitive Psychology. New York: Harper & Row Publishers, 1982.

Winograd, T. On some contested suppositions of generative linguistics about the scientific study of language. Cognition, 1977, 5, 151-179.

Witkin, H., Moore, C. A., Goodenough, D. & Cox, P. Field dependent and field-independent cognitive styles and their educational implications. Review of Educational Research, 1977, 47(1), 1-64.